Institut de recherche
pour le développement

Délégation aux Systèmes d'Information
Pôle des "services d'appui à la recherche"
Informatique scientifique

Appel à projets interne

SPIRALES 2009

Formulaire de demande DSI-SPIRALES

« Soutien aux Projets Informatiques dans les Equipes Scientifiques »

2nd cas: Nouveau projet (projet finalisé de développement d'une application IS: base de données, application web , algorithmie…)

Remise des projets :

14 novembre 2008

à spirales@ird.fr

Contact :

Régis Hocdé - Informatique Scientifique

regis.hocde@ird.fr

ou equipe-is@ird.fr

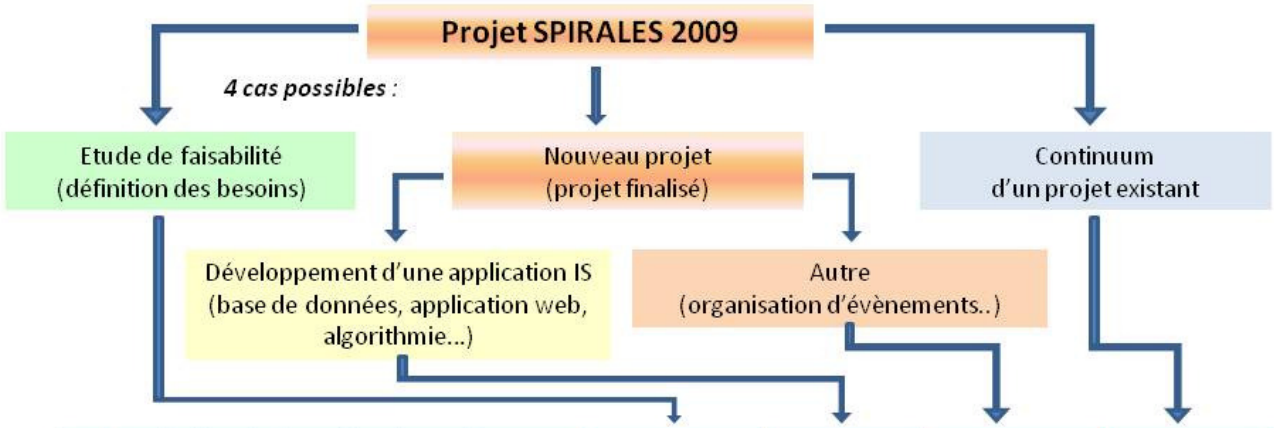
Demande d'un soutien DSI sur les projets informatiques des UR/US.

Formulaire de demande DSI-SPIRALES 2009
« Soutien aux Projets Informatiques dans les Equipes Scientifiques ».

Le présent formulaire comporte différentes parties, avec des questions parfois très précises. Il est destiné à **faciliter le travail des évaluateurs**.

Les propositions doivent être adressée sous forme électronique (au format RTF, DOC ou PDF) à l'adresse suivante : spirales@ird.fr

4 cas sont possibles dans le cas d'un projet SPIRALES 2009.


Projet SPIRALES 2009
4 cas possibles :
Etude de faisabilité (définition des besoins)
Nouveau projet (projet finalisé)
Continuum d'un projet existant
Développement d'une application IS (base de données, application web, algorithmie...)
Autre (organisation d'évènements..)

	Parties du formulaire à renseigner selon le cas	*1er cas*	*2nd cas*	*3ème cas*	*4ème cas*
1.	Nature du projet	Oui	Oui	Oui	Oui
2.	Porteur(s) de projet	Oui	Oui	Oui	Oui
3.	Moyens / appui demandés à la DSI	Oui	Oui	Oui	Oui
4.	Bilan / Etat d'avancement des phases précédentes	Non	Non	Non	Oui
5.	Description des besoins	Oui	Oui	Oui	Non
6.	Description du projet de développement d'application IS (méthodes, solutions, et moyens)	Non	Oui	Non	Oui (*)
7.	Description du projet hors développement (méthodes, solutions, et moyens)	Non	Non	Oui	Non
8.	Pertinence, résultats/livrables attendus et valorisation du projet	Oui	Oui	Oui	Oui

(*) Dans le cas d'un **continuum**, si le formulaire 2008 a déjà été détaillé, la **partie 6 peut être remplacée par tous documents utiles**: dossier d'expression des besoins ou cahier des charges, devis détaillé…

Ce formulaire est spécifique au <u>2nd cas: Nouveau projet (projet finalisé de développement d'une application IS: base de données, application web , algorithmie…).</u>

1. Nature du projet

1.1. Titre du projet

Visual_YAO : Réalisation d'une version graphique de YAO, outil logiciel pour l'assimilation variationnelle des données dans les modèles numériques.

1.2. Résumé du projet proposé (5 lignes maximum)

Actuellement YAO est un logiciel en mode texte utilisé avec succès par des géophysiciens possédant une bonne maîtrise de la programmation, pour faire de l'Assimilation Variationnelle de Données. Ce projet vise à élaborer une version graphique de YAO, plus aisée d'appropriation et d'utilisation par le non informaticien. Cela permettrait notamment de répondre aux problématiques de recherche de la JEAI/MTE (LTI/ESP) utilisant l'inversion neuro-variationnelle basé sur YAO pour l'estimation de la chlorophylle-a au large du Sénégal. Une version graphique de YAO permettrait également sa diffusion auprès d'une plus grande communauté scientifique.

1.3. Type de projet

<table>
<tr><td>1er cas</td><td colspan="2">· <u>Etude de faisabilité (définition des besoins)</u> : Demande d'appui pour une analyse fine des besoins et la formulation de spécifications, éventuellement développement d'un prototype (en vue d'une seconde phase destinée au développement et à la réalisation du projet)</td></tr>
<tr><td>2nd cas</td><td rowspan="2"><u>Nouveau projet</u> SPIRALES :</td><td>☑ <u>Projet finalisé de développement d'une application IS</u> (proposition finalisée et détaillée en matière d'expression des besoins, d'identification des solutions et des moyens…),</td></tr>
<tr><td>3ème cas</td><td>· Projet <u>autre qu'un développement</u> d'application IS (proposition finalisée concernant tous autres domaines : animations, évènements, traitement de données, calcul intensif…),</td></tr>
<tr><td>4ème cas</td><td colspan="2">· <u>Continuum</u> d'un projet SPIRALES existant (prévu sur 2008-2009 ou suite d'un précédent projet SPIRALES)
Continuum d'un <u>développement d'une application IS</u>
o continuum d'un projet <u>autre qu'un développement</u> d'application IS</td></tr>
</table>

Les demandes d'hébergement d'applications IS, d'accès à un serveur de développement, de création de dépôt Subversion (SVN), de formations IS…
ne constituent pas des demandes SPIRALES et doivent être adressées directement à equipe-is@ird.fr sans échéance particulière.

1.4. Durée prévue :

☐ Durée prévue : ☐ 1 an ☒ 2 ans

Pour les continuums : date de démarrage du projet : ▪ 2004 ▪ 2005 ▪ 2006 ▪ 2007 ▪ 2008

2. Porteur(s) de projet

2.1. Unité :

☒ UMR ☐ UR ☐ US N° Nom : LOCEAN

2.2. Département

☒ DME ☐ DRV ☐ DSS

2.3. Nom du porteur de projet :

Nom prénom : NIANG Awa

2.4. Statut et coordonnées du porteur de projet :

Statut / Catégorie – Localisation géographique – Téléphone – Fax – E-mail

Maître de Conférences. LOCEAN, 4, Place Jussieu (UPMC), barre 45-55 5ème étage. Paris. Tel +33 1 44 27 27 08, Fax : +33 1 44 27 71 59 E-mail : awa@locean-ipsl.upmc.fr

LTI , Ecole Supérieure Polytechnique (ESP), Université de Dakar, Tel : +221 77 642 55 16, Fax : +221 33 825 61 64

2.5. Nom et coordonnées du Directeur d'Unité (si différent) :

Nom prénom - Statut / Catégorie – Localisation géographique – Téléphone – Fax – E-mail

Laurence EYMARD D.R. CNRS ; LOCEAN 4, Place Jussieu (UPMC), barre 45-55 4ème étage, pièce 407 Paris. Tel : +33 1 44 27 70 73 Fax : +33 1 44 27 38 05 E-mail : laurence.eymard@locean-ipsl.upmc.fr

2.6. Aval du directeur d'unité (obligatoire).

OU adressé par mail à spirales@ird.fr

Pour accord

Laurence Eymard

2.7. Implantation principale de l'unité :

LOCEAN. 4 Place Jussieu, Tour 45-55, Boîte 100 75252 Paris cedex 05

2.8. Site(s) de déroulement du projet :

Laboratoire de Traitement de l'Information (LTI) ESP/UCAD Dakar

LOCEAN/UPMC/Paris

2.9. Site administratif à partir duquel se feront les dépenses budgétaires

(*attention: cette information doit être bien renseignée car elle évite les disfonctionnements observés quand il s'agit de transférer les crédits d'un site à un autre en cours d'année*)

Représentation IRD de Dakar Sénégal

2.10. Projets inter-unité ou inter-organismes :

☒ **Projet inter-unités** ☒ **Projet inter-organismes**

2.11. Liste des unités ou organismes partenaires du projet

Organisme – Unité – Directeur d'Unité - Localisation géographique *(autant de fois que nécessaire)*

- UCAD Laboratoire de Traitement de l'Information, ESP/UCAD, Claude LISHOU, Dakar
- CEA-CNRS-UVSQ Laboratoire des Sciences du Climat et de l'Environnement (LSCE), UMR CEA-CNRS-UVSQ, Robert VAUTARD**,** Orme des Merisiers, Gif-Sur-Yvette

2.12. Liste des intervenants impliqués de manière effective dans la réalisation du projet :

(autant de fois que nécessaire)

Nom prénom - Statut / Catégorie – Unité - Localisation géographique – Contribution envisagée

Awa Niang, MC, LOCEAN, UPMC/Paris et LTI/Dakar, Modélisateur, Conception -Encadrement

Roger Faye, MC, LTI/Dakar Modélisateur, Conception - Encadrement

Alex Corenthin, MA LTI/Dakar, Informaticien Conception - encadrement

Carlos Mejia, IR CNRS, LOCEAN, UPMC/ PARIS, Informaticien Conception - encadrement

Luigi Nardi Thésitif , LOCEAN, UPMC/ PARIS, Informaticien Conception

Sylvie Thiria, PU, LOCEAN, UPMC/ PARIS, Modélisateur - Encadrement

Cyril Moulin, IR CEA, LSCE-CEA/SACLAY Géophysicien -Validation- Encadrement

Abdou Kane, Thésitif, LSCE-CEA/SACLAY Géophysicien -Validation

2.13. Disponibilité / implication de chacun des intervenants effectifs : exprimée en % de temps-homme ou en jours-homme (ETP total ou pour une période)

Ex : Dupont – forte disponibilité - 50% ETP sur la durée du projet

Martin – très faible disponibilité – 0,5 jour / mois

Awa Niang –forte disponibilité- 50% ETP sur la durée du projet

Alex Corenthin - bonne disponibilité- 15% ETP sur la durée du projet

Roger Faye – Moyenne disponibilité- 10% ETP sur la durée du projet

Carlos Mejia – bonne disponibilité- 15% ETP sur la durée du projet

Luigi Nardi – bonne disponibilité- 15% ETP sur la durée du projet

Sylvie Thiria –Faible disponibilité- 1 jour par mois

Cyril Moulin Faible disponibilité- 0,5 jour par mois

Abdou Kane – Moyenne disponibilité- 10% ETP sur la durée du projet

L'essentiel est de donner un ordre de grandeur (et non pas une évaluation monétaire) : s'agit-il de 4 jours de travail (4 jours ETP) pour l'année, 15 jours ETP ou 40 jours ETP (un jour par semaine) ou de s'impliquer à temps complet (200 jours ETP)… ?

3. Moyens / appui demandés à la DSI

3.1 Contribution financière demandée à la DSI pour 2009 en euros HT :

Montant 2009 demandé : 16500€ HT soit

Ventilation par poste :

Fonctionnement : 6000€

Equipement : 4000 €

Prestation de service : 6500€ (4000€ ingénieur de développement, 2x1250 stage élève ingénieur)

(les informations apportées doivent être cohérentes avec celles précisées à la question 24.)

3.2 Demande envisagée pour 2010 – si projet de 2 ans - en euros HT :

Montant 2010 envisagé : 12000 € soit

Ventilation par poste :

Fonctionnement : 6000€

Equipement :

Prestation de service : 6000€ ingénieur de développement

3.3 Montant(s) précédemment attribué(s) par la DSI - en euros HT :

	2004	2005	2006	2007	2008
Montants attribués (€ HT)					

3.4 Moyens affectés au projet et Cofinancements acquis hors SPIRALES (€ HT) :

Autres sources de financements acquis :

JEAI MTE « Modélisation et Traitement de Données Environnementales » / ESP/UCAD : environ 9000 euros en 2009

LSCE : projet TOSCA bourse de thèse Nardi Montant (32000€ par an HT) :

Moyens apportés par l'unité (hors ressources humaines) :

- 63 250,00 € HT Montant (€ HT) par le Service Hydrographique et Océanographique de la Marine (SHOM). Contrat de Recherche N° 08 CR 0002 : « Système d'inversion acoustique et océanographique basée sur l'assimilation de données (SINOBAD) ;

- Moyens informatiques, fournitures.

3.5 Moyens humains affectés au projet :

Total des moyens humains affectés au projet par les unités et partenaires (exprimé en total de jours-homme ou ETP (Equivalent Temps Plein)) (*cf. définition et exemple à la question 17.*) :

(Awa Niang 50% , Alex Corenthin 15% , Roger Faye 10%, Carlos Mejia 15%, Luigi Nardi 15% , Sylvie Thiria 1 jour par mois, Cyril Moulin 0,5 jour par mois, Abdou Kane 10% , Elève ingénieurs 240%, Prestation de service 50%),

Soit 823 jours- homme

3.6 Coût total estimé du projet (toutes années confondues) :

Estimation du coût total du projet toutes années SPIRALES confondues : crédits SPIRALES, moyens fournis par l'unité et cofinancements acquis (hors ressources humaines) : 132 000 € HT

3.7 Ressources humaines extérieures mobilisées ou demandées:

☐ Compétences mobilisées ou souhaitées (profil type) :

☒ Intervention d'un/de prestataire(s) de service : Ingénieur Informatique

☒ Mobilisation d'un/de stagiaire(s) *(sous réserve de compétences fortes en informatique scientifique au sein de l'équipe porteur du projet et de capacités de l'équipe à dégager du temps pour assurer un réel encadrement)*

2 Elèves Ingénieurs de Département Informatique de l'ESP/UCAD. Co-encadrés par Alex Corenthin, Roger Faye Awa Niang et Carlos Meijia.

☐ Demande d'appui de l'équipe 'Informatique scientifique' de la DSI / pour l'appui méthodologique et le suivi de projet :

Non

☐ Demande d'appui de l'équipe 'Informatique scientifique' de la DSI / pour le développement et/ou la réalisation du projet (avec estimation du temps-homme nécessaire) : Non

*La DSI, suite au comité d'évaluation, pourra pour quelques projets et sur quelques sites (Nouméa, Dakar, Montpellier…) et dans la limite des **moyens humains de la DSI disponibles**, convertir ces demandes d'appui ou de financement de prestataire de service en temps-homme, c'est-à-dire par une intervention directe du 'pool informatique scientifique'.*

3.8 Demande d'un dépôt Subversion (SVN) :

Description des besoins pour ce projet SPIRALES (une demande formelle et détaillée, avec signature de la charte sera néanmoins nécessaire dans un 2nd temps) - (Définition SVN: http://fr.wikipedia.org/wiki/Subversion_(logiciel))

3.9 Demande d'hébergement(s) / d'accès à un (des) serveur(s) 1/ de développement et de tests pour la durée du projet, 2/ de 'pré production' et de recette pendant ou à l'issue du projet, 3/ d'exploitation à l'issue du projet :

Description des besoins pour ce projet SPIRALES : technologies, capacité… (une demande formelle et détaillée, avec signature de la charte sera néanmoins nécessaire dans un 2nd temps)

Non

3.10 Appui de la DSI apporté pour l'élaboration du projet ?

Si vous avez bénéficié de l'appui de la DSI (coordination IS, pool d'informaticiens scientifiques de Dakar ou Nouméa, SIL…) pour l'élaboration de cette proposition, décrivez très brièvement le type d'appui.

Non

5. Description des besoins

La demande peut-être être **accompagnée de tous documents utiles** :

*présentation du projet global ou descriptif du projet, rapport de phases préliminaires, étude de faisabilité (définition des besoins), dossier d'expression des besoins ou **cahier des charges**, devis détaillé…*

5.1 Objectifs scientifiques

Actions ou projets de recherche soutendus par ce projet SPIRALES

OU renvoyer à un document joint

Voir document joint

5.2 Description et analyse des besoins

OU renvoyer à un document joint

Voir document joint

5.3 Description de l'existant (moyens – outils – compétences)

OU renvoyer à un document joint

Voir document joint

5.4 Difficultés rencontrées jusqu'à présent :

Voir document joint

6. Description du projet – SEULEMENT SI « développement d'application IS » (méthodes, solutions, et moyens)

La demande peut-être **accompagnée de tous documents utiles** :

*présentation du projet global ou descriptif du projet, rapport de phases préliminaires, étude de faisabilité (définition des besoins), dossier d'expression des besoins ou **cahier des charges**, devis détaillé…*

6.1. Nom de votre outil

Visual_YAO

6.2. Si votre outil existe déjà, quel est l'URL du site internet ou des documents qui le décrivent? Ou, si l'outil a été décrit dans un article, fournir les références

Ci dessous l'URL pour l'outil YAO (version textuelle).

http://www.ipsl.jussieu.fr/groupes/NEURATEL/documents/Yao/YaoAbstract.htm#RapportYao

1. Innovation :

6.3. Ecrire 3 scénarios qui illustrent comment votre outil sera ou a été utilisé dans votre communauté scientifique ou domaine d'activités

1) Développement d'une nouvelle application d'assimilation de données avec Visual_YAO. L'utilisateur dispose d'une icône pour cela. Il y aura une fenêtre lui permettant la création d'une nouvelle application avec possibilité de visualiser le graphe.

2) On peut également reprendre une application physique comme NEMO par exemple et la modifier avec Visual_Yao, pour par exemple visualiser son graphe, ce que l'on ne pouvait pas faire avec YAO. On peut également

3) Changer un fichier d'instruction : se resservir d'une application mais en changeant seulement le fichier d'instruction.

6.4. Décrire, en un paragraphe, les innovations de votre projet pour votre communauté scientifique

Dans l'état actuel l'application est lancée directement à partir de la ligne de commande et les résultats affichés dans la même fenêtre sous forme de trace peu lisible. L'interface graphique devrait permettre l'exécution et la visualisation des résultats de façon claire. Enfin, l'interface graphique faciliterait grandement l'utilisation de YAO et donc la mise en oeuvre de l'assimilation de données pour les non informaticiens.

L'analyse d'erreur dans une application de grande dimension avec YAO se révèle très complexe (on ne sait pas où se trouve l'erreur et sur quel attribut il faut opérer). Cette complexité peut être largement réduite si on utilise VISUAL_YAO qui aura une interface graphique qui contrôlera la syntaxe à l'écriture, mettra immédiatement en évidence l'incohérence des attributs et ne permettra pas la continuation de la définition de l'application.

Une visualisation du graphe modulaire peut elle aussi s'avérer une aide importante pour la compréhension de l'application et doit en conséquence intéresser un grand nombre de scientifiques, utilisateurs potentiels de YAO. Le but est d'avoir un I.D.E. (Interface Development Environment) pour le développement des applications. Cette interface graphique accompagnera l'utilisateur dès la création de l'application jusqu'aux tests et à la vérification des résultats.

6.5. Existent-ils d'autres outils similaires au vôtre ? Si c'est le cas, lister ces outils et décrire les avantages de votre outil par rapport aux autres

Non

6.6. Si vous proposez des améliorations à un outil existant, combien d'utilisateurs ont déjà téléchargés ou obtenus une copie de la version actuelle ?

Plusieurs utilisateurs ont obtenus et utilisé une version YAO: CLS, ULB, LSCE, CETP, SHOM.

6.7. Le projet proposé est-il basé sur de nouvelles conclusions scientifiques ou méthodes innovantes ? Si c'est le cas, décrire les fondements et lister les références les plus pertinentes.

Non

2. Calendrier, budget et risques

6.8. Calendrier du projet montrant les tâches clés et les dates d'échéances

1 mois de formation sur YAO en février 2009.

9 mois de développement du noyau base de l'architecture graphique, c'est-à-dire développement du pattern MVC (Model View Control) et premières boîtes de dialogue (création d'un projet, définition des toutes les directives YAO sauf celle des modules pour la description du graphe modulaire).
6 mois pour créer un éditeur graphique dédié à la création du graphe modulaire et au chargement des anciens projets.
2 mois pour développer la définition des instructions propres au processus d'assimilation de données.
2 mois pour développer le chargement et modification d'un projet YAO existant.
2 mois pour créer la documentation sur la nouvelle interface et sur l'aide en ligne de Visual_YAO.
1 mois d'étude sur l'ergonomie de l'interface.
1 mois de test et validation des résultats par les utilisateurs.

6.9. Eventuellement, budget détaillé montrant les coûts des tâches clés, des différents modules ou phases.

(*Les informations apportées doivent être cohérentes avec celles précisées à la question 18.*)

6.10. Si vous demandez des fonds pour des activités autres que du « développement logiciel », pourquoi ces activités sont-elles essentielles à l'accomplissement de votre projet ?

Il sera nécessaire d'effectuer des missions entre Paris et Dakar dans les deux sens, notamment pour la formation et l'encadrement.

6.11. Quel sont les risques encourus si votre projet ne peut être finalisé à échéance et dans le budget prévu ? Comment comptez-vous pallier à ces risques ?

La conception de l'interface graphique sera faite de façon à obtenir comme premiers résultats une version minimale déjà utilisable. Au fur et à mesure des tâches viendront s'ajouter. Ne pas finaliser notre projet à échéance veut dire ne pas développer toutes les tâches mais avoir, en tout cas, une application déjà aboutie. Il n'y a pas, donc, des risques énormes à pallier.

6.12. Si vous demandez un soutien d'un an, accepteriez-vous de recevoir les crédits l'année prochaine plutôt que cette année ?

Non

6.13. Si cette demande concerne la phase 1 d'un projet prévu sur 2 ans, pouvez-vous réaliser le projet en entier sur une année si vous obtenez les crédits en une seule fois ? Comment cela impacterait-il votre projet ?

Non

3. Architecture de l'outil

6.14. Décrire l'architecture envisagée pour votre outil. Identifier les composants clés de l'application et décrire comment ils interagissent.

Le Modèle-Vue-Contrôleur (en abrégé MVC) est une architecture et une méthode de conception qui organise l'interface Homme-machine d'une application logicielle. Il divise l'ihm en un modèle (modèle de données), une vue (présentation, interface utilisateur) et un contrôleur (logique de contrôle, gestion des événements, synchronisation), chacun ayant un rôle précis dans l'interface (figure 1).

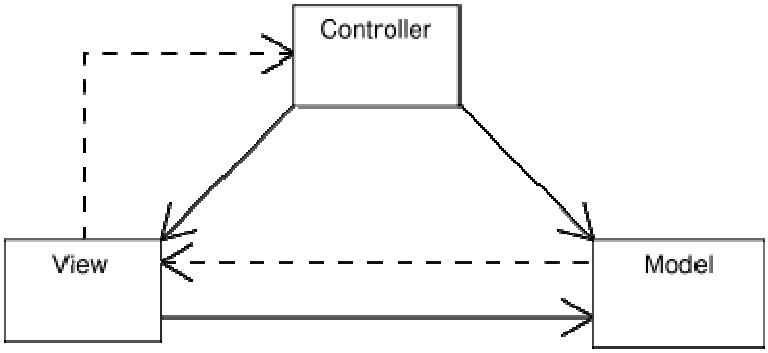


Figure 1 : Architecture Modèle-Vue-Contrôleur

Ce pattern rend plus aisé l'intégration de Visual_YAO avec le YAO textuel et donc l'architecture déjà existante. Les interactions parmi les composants sont celles typiques du pattern MVC. Le composant clé de l'architecture à développer est le composant Vue car le Contrôleur et le Modèle existent déjà et doivent être adaptés.

Nous présentons ci-dessous (figure 2) le schéma de procédure de YAO.

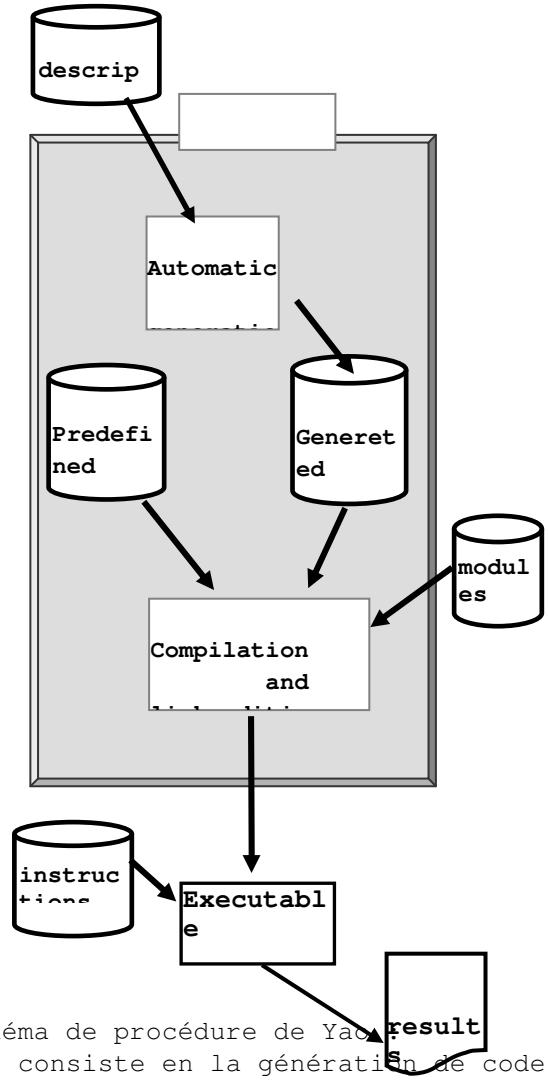


figure 2 : schéma de procédure de Yao. Une première étape consiste en la génération de code à partir du fichier de description. Puis ce code généré est intégré au code prédéfini de Yao. L'ensemble est ensuite compilé et linké pour produire un exécutable qui saura finalement interpréter des instructions (en batch ou en interactif) pour produire un résultat.

6.15. Lister les méthodes/référentiels d'analyses, de conception et de développement utilisés pour élaborer l'outil.

- UML comme langage de modélisation ;
- Principes interface Homme-machine pour l'ergonomie de l'interface.

6.16. Lister les langages de programmations et les outils de développement envisagés. Préciser le type de syntaxe qui sera utilisée pour la documentation du code.

- Langage de programmation C++ ;
- wxWidget pour le développement de l'interface graphique en C++ ;
- Doxygen : outil pour la génération de la documentation du code (API).

6.17. Lister le matériel et les logiciels requis pour faire fonctionner votre outil.

- Système d'exploitation LINUX (avec compilateur g++) ;
- Version actuellle de l'outil YAO.

6.18. Comment ces choix influeront sur l'appropriation de votre outil par les utilisateurs cibles ?

- Les choix effectués ne présentent aucune difficulté pour l'appropriation de l'outil par les utilisateurs et seront transparents lors de l'utilisation.

6.19. Justifier le choix de ces technologies (conformité à des référentiels, robustesse, pérennité, communauté de développeur importante…) :

- Le système d'exploitation et les logiciels utilisés sont libres, faciles à obtenir. De plus ils sont robustes et pérennes et sont très largement utilisés dans la communauté scientifique.
- wxWidget permet de développer des applications graphiques en C++. Il permet aussi de faciliter la compatibilité et la synchronisation entre le « Presentation tier » et le « Logic tier » au moment de l'union du nouveau code avec l'ancien car ils seront écrits dans le même langage. Il y a aussi une forte réutilisation du code déjà écrit puisque chaque directive YAO a une correspondance univoque avec une classe dans le code YAO existant et donc on retrouve déjà les structures de données pour stocker les entrées de l'interface graphique.
- Doxygen permet de générer automatiquement la documentation à partir des commentaires dans le code.

4. Données en entrée et en sortie

6.20. Énumérer et décrire les données en entrée et en sortie de votre outil.

- En entrée on doit spécifier le modèle direct de l'application étudiée et les données à assimiler ;
- En sortie l'outil fournit des fichiers de spécification de l'application et un exécutable permettant d'effectuer l'assimilation variationnelle de données dans l'application mise en entrée. Les résultats de cette assimilation sont également visualisables.

6.21. Décrire la disponibilité (ou l'accessibilité), le format de stockage et d'organisation ainsi que la qualité des données utilisées en entrée. Quel est le coût et l'effort requis de l'utilisateur pour collecter, acheter, pener ou convertir ces données ? Dans quelles mesures le coût et l'effort requis limiteront-ils l'adoption de votre outil ?

Cet outil graphique est conçu pour permettre à la communauté des physiciens et géophysiciens d'aborder plus facilement l'assimilation variationnelle des données dans leurs modèles. L'effort requis en utilisant l'outil Visual_YAO sera moindre que si on utilise une autre méthode.

6.22. Les données seront-elles testées ou validées par l'outil en entrée ? Si oui, comment ?

Un des buts de Visual_YAO est de permettre un contrôle plus strict de la spécification du modèle direct de l'application étudiée.

En effet la syntaxe à l'écriture met immédiatement en évidence l'incohérence des attributs et ne permet pas la continuation de la définition de l'application.

6.23. Validerez-vous ou avez-vous déjà validé scientifiquement les données en sortie de votre outil ? Si oui, décrire comment cela se fera ou a été fait.

Les programmes créés par Visual-YAO seront validés en contrôlant les fichiers de spécification créés dans YAO. Le fonctionnement de YAO a déjà fait l'objet d'une validation.

6.24. Décrire l'utilité immédiate des données en sortie de votre outil et les nécessaires conversions, post-traitements ou analyses ultérieures requis. Comment l'effort requis impactera-t-il l'adoption de votre outil par les utilisateurs cibles ?

Réalisation simplifiée d'une assimilation de données.

6.25. Existent-ils des métadonnées ou y a-t-il production de métadonnées décrivant les lots de données en entrée ou sortie ? Si oui, comment sont-elles gérées et entreposées ? Sont-elles basées sur des standards ?

Non

6.26. La description ou le référencement des données est-il / sera-t-il basé sur un ou des référentiels ou thésaurus ? Si oui, lesquels ?

Non.

5. Interopérabilité

6.27. Quels sont les éventuels standards ou normes utilisées ?

Interface Homme-machine, UML et écriture des commentaires, à la façon javadoc, pour la génération des API avec Doxigen.

6.28. Votre outil est-il prévu pour être utilisé de manière interactive par les utilisateurs, par d'autres outils ou programmes (communication entre outils sur la base de requêtes ou autres) ou les deux ?

Oui l'interface graphique est une interface Homme-machine donc interactive. Pourtant une interaction avec d'autre outils ou programmes n'est pas encore prévue.

6.29. Si votre outil pourra être utilisé dans les 2 cas, de manière interactive et de manière automatisée par d'autres applications, décrire les caractéristiques et fonctionnalités non accessibles pour chaque mode d'utilisation.

6.30. Si votre outil pourra communiquer de manière automatisée avec d'autres programmes,

écrire brièvement 3 scénarios d'utilisation qui illustrent les détails de ces communications.

6.31. Si votre outil intégrera ou fera appel à des outils d'autres développeurs, décrire brièvement 3 scénarios d'utilisation

- L'outil prévoit l'utilisation de deux minimiseurs M1QN3 et M2QN1 développés à l'INRIA. L'utilisateur doit au préalable faire une démarche d'autorisation auprès de l'INRIA. Sinon d'autres minimiseurs sont fournies dans l'outil.

6. Rapports d'erreurs et d'avancement

6.32. De quelle manière votre outil montrera la progression du traitement aux utilisateurs ? Qu'est-ce qui sera signalé ?

Dans la phase de déclaration du modèle direct, la progression de l'utilisateur sera empêchée s'il est en train de spécifier des directives incohérentes par rapport aux directives déjà spécifiées.
Dans la phase d'exécution du processus d'assimilation de données on aura une barre de progression qui visualisera l'état d'avancement en fonction du numéro d'itération propre à l'assimilation de données.

6.33. Comment votre outil notifiera-t-il à l'utilisateur l'apparition d'une erreur et quelles informations seront affichées dans le message d'erreur ?

Dans la phase de déclaration du modèle l'empêchement à continuer visualisera un message d'erreur décrivant l'incohérence trouvée.

Des messages d'erreurs complétés d'un code d'erreurs pour la phase d'assimilation.

6.34. Avez-vous mis en place un processus de gestion des erreurs et de correction par l'équipe de développement et comment ?

Deux personnes de l'équipe projet (C. Moulin et A. Kane) auront pour rôle essentiel d'effectuer des tests à différents moments du développement et pourront alors échanger avec l'équipe de développement pour les corrections éventuelles.

7. Documentation

6.35. Quelles sont les différentes documentations prévues : nature et format de la (des) documentation(s) ? cible visée ? (spécifications fonctionnelles, spécifications techniques, docs/API développeurs…)

- Aide en ligne ;
- API développeur à générer avec Doxigen ;
- Tutorial de formation ;
- Exemples d'utilisation ;
- Publications associées ;
- Manuel d'utilisation.

6.36. Lister les sujets ou principaux chapitres qui apparaitront dans la/les documentation(s) de votre outil

L'aide en ligne aura une section pour chaque directive YAO, une section pour la création et la modification d'un projet YAO et une section pour la spécification des instructions et l'analyse des résultats de l'assimilation de données.

Les API seront standard : interface de programmation que les classes de Visual_YAO mettent à disposition.

Le tutorial de formation sera le rapport déjà existant de YAO modifié en ajoutant les nouvelles caractéristiques de Visual_YAO.

Les exemples d'utilisation sont des applications développées avec Visual_YAO et qui peuvent être prises comme exemple pour les utilisateurs futurs. Il y aura plusieurs exemples présentant chacun les diverses phases de réalisation d'une application d'assimilation des données avec Visual_YAO.

Les publications de YAO textuel seront encore valables car les expériences d'assimilation de données pourront être exécutées à nouveau avec Visual_YAO. De nouvelles applications d'assimilation de données avec le nouvel outil pourront certainement faire l'objet de publications.

Le manuel d'utilisation sera un document plus complexe. Il sera l'équivalent de l'aide en ligne mais beaucoup plus riche et détaillé. Il sera en résumé composé des chapitres suivants :

- Généralités ;
- Installation de Visual_YAO;
- Création d'un projet et modification d'un projet existant;
- Spécification des directives YAO : lexique, syntaxe et sémantique ;
- Définition des instructions pour piloter le processus d'assimilation ;
- Exécution et analyse des résultats ;
- Exemples de spécification des directives YAO.

8. Multilinguisme - traduction

6.37. Lister les langues parlées par vos utilisateurs cibles.

Français et anglais.

6.38. Lister les langues dans lesquelles votre outil, votre documentation et tous les autres livrables seront traduits. Si vous ne traduisez par votre outil dans toutes les langues parlées par vos utilisateurs, comment cela affectera-t-il l'adoption de votre outil ?

Toute la documentation sera en français et en anglais. Le logiciel est pour l'instant adressé aux francophones. Une partie de la documentation sera envisagée en anglais en prévision d'une internationalisation de l'outil. L'anglais ne posera pas de problème aux francophones, il n'y aura pas d'affectation négative.

6.39. Quelles méthodes ou technologies seront utilisées pour la traduction de votre outil, votre documentation et des autres livrables?

Tous les outils de traduction automatique seront bien acceptés aussi bien que des correcteurs automatiques, comme « aspell » par exemple.

Processus et équipe de développement

6.40. Avez-vous déjà géré des projets de développement logiciel précédemment ? Décrire brièvement votre(vos) expérience(s) passée(s).

Certains membres de l'équipe du projet ont développé le logiciel YAO. Son développement a duré plusieurs années et a mobilisé entre autres un ingénieur informaticien à temps complet. D'autre part plusieurs membres de l'équipe projet sont

informaticiens et sont donc familiers des techniques utilisés en génie logiciel.

6.41. Les développements seront-ils réalisés par des membres de votre équipe, par un prestataire sous contrat, ou autre ?

Au début du projet les développements seront réalisés en partie par les élèves ingénieurs encadrés dans le projet et par certains membres de l'équipe. Nous prévoyons également d'employer un étudiant en Informatique comme prestataire de service.

6.42. Si vous avez déjà sélectionné des développeurs, de votre équipe ou d'un prestataire, lister les, spécifier leurs rôles et décrire leurs compétences et leurs expériences passées. Attacher leurs CV si vous les avez.

6.43. Si vous envisagez un prestataire de service, avez-vous déjà travaillé avec un prestataire auparavant ? Décrire comment vous vous assurerez qu'il développe ce que vous recherchez, dans les temps et avec le budget prévu.

Nous avons déjà travaillé avec des prestataires de services (étudiants sous contrat) à l'occasion d'autres projets. Le suivi du développement sera régulier (réunions bimensuelles) pour une présentation du travail réalisé avec démonstration et documentation exigée.

6.44. Impliquerez-vous vos utilisateurs cibles dans le processus de conception et d'implémentation de l'outil ? Si oui, décrire comment.

Au sein du LOCEAN il y a plusieurs chercheurs qui travaillent constamment avec le YAO textuel. Ils vont suivre les développements envisagés et seront sollicités pendant la phase de conception et de test. En plus, les formations qui sont envisagées dans le projet surtout à Dakar, en particulier pour les membres de la JEAI/MTE constitueront un retour sur expérience qui nous permettra d'améliorer l'outil et sa convivialité.

6.45. Où sera hébergé le code source de votre outil durant son développement puis durant sa maintenance ?

L'outil sera développé conjointement au LTI et au LOCEAN. Il sera hébergé au LOCEAN.

6.46. L'outil sera-t-il placé dans une plateforme collaborative ou au sein d'une communauté de développement de projets open-source ? si oui, lesquels ?

Nous n'envisageons pas le placement de Visual_YAO sur une plate-forme collaborative comme source-forge. Cependant, cette option n'est pas à écarter si la communauté autour du projet devient suffisamment grande. L'outil sera disponible au LOCEAN à tous les utilisateurs du réseau.

9. Licence et distribution

6.47. L'utilisation de l'outil sera-t-elle soumise à une licence pour les utilisateurs qui l'installeront sur leurs propres machines? S'agira-t-il d'une licence libre ? Le code source de l'outil sera-t-il protégé ou complètement ouvert ? (décrire l'éventuel coût, le type de licence et toutes autres éventuelles obligations)

Visual_YAO sera Open-source en exclusion de la partie des minimiseurs M1QN3 et M2QN1.

6.48. Existe-t-il des parties ou modules de votre outil qui sont protégés par des brevets ou des marques ?

La librairie de M1QN3 et M2QN1 est propriété de l'INRIA, elle peut être incluse dans Visual_YAO sous acceptation de cette

organisation.

6.49. Décrire comment l'outil sera distribué ou rendu accessible aux utilisateurs (lister les sites web si nécessaire)

L'outil sera téléchargeable directement sur internet, sur le site du groupe NEURATEL http://www.ipsl.jussieu.fr/groupes/NEURATEL/

10. Installation

6.50. La procédure d'installation sera-t-elle automatisée par un programme ou un script ou l'outil devra-t-il être installé « manuellement » ? (Préciser les OS et distribution)

On envisage d'utiliser les AUTO TOOL pour avoir une procédure d'installation automatique et standard pour tous les systèmes UNIX-like. Cette procédure automatique devrait prendre en compte le type d'architecture et générer un script de configuration qui viendra s'intégré au makefile d'installation paramétrique et fonction de la configuration. Cette méthode nous donnera une installation indépendante de l'architecture.

6.51. Est-ce que le programme ou script d'installation détectera et signalera les logiciels requis manquants ?

Oui

6.52. Est-ce que le programme ou script d'installation permettra la désinstallation de l'outil ?

Oui

6.53. Si l'installation n'est pas pris en charge par un programme ou un script, existera-t-il une notice d'installation ?

6.54. De quelle manière la complexité de la procédure d'installation limitera l'adoption/l'utilisation de l'outil par les utilisateurs cibles ?

La complexité est minimale car l'installation sera faite comme pour beaucoup d'autres programmes.

11. Opération

6.55. Les utilisateurs pourront-ils faire fonctionner l'outil sans votre aide? Si les utilisateurs doivent solliciter votre équipe ou des consultants externes ou suivre une formation, décrire les détails et les coûts

L'interface graphique est une aide à l'assimilation de données faite avec YAO. Pour s'en servir il faut en tout cas connaître les principes de l'assimilation de données et la façon dont YAO gère l'assimilation. A partir de ces pré requis, l'outil pourra être utilisé sans l'aide de l'équipe de projet car le projet Visual_YAO est destiné à des non informaticiens.

12. Assurance qualité, maintenance et support

6.56. Lister les techniques que votre équipe utilisera pour détecter les erreurs ou défauts.

Nous utiliserons les techniques classiques de débogage.

6.57. Dans le cas où vous auriez un programme 'beta' en fin de développement, décrire comment il fonctionnera. Si des utilisateurs se sont déjà engagés pour l'utiliser, listez-les.

6.58. De quelle manière votre équipe fera-t-elle le suivi des erreurs dans ce projet ?

L'outil YAO, et puis Visual_YAO, aura toujours un responsable qui sera chargé du suivi des erreurs.

6.59. De quelle manière apporterez-vous un support à vos utilisateurs pendant la durée de ce projet, et après ?

On est toujours très ouverts aux collaborations avec les nouveaux utilisateurs de YAO, on le sera également avec les utilisateurs de Visual_YAO. Pendant la durée du projet Visual_YAO ne sera pas encore opérationnel, les utilisateurs seront obligés d'utiliser la version textuelle de YAO. Les utilisateurs actuels de YAO (textuel) peuvent et pourront nous contacter au travers de courriel.

6.60. De quelle manière apporterez-vous un appui aux développeurs d'autres outils qui souhaiteraient utiliser et intégrer votre outil aux leurs ?

Pour ceux qui veulent se servir de Visual_YAO pour le développement d'autres outils, ils auront le maximum de disponibilité de notre part. On laissera télécharger le logiciel et la documentation. Des formations sur l'assimilation avec YAO sont également faites quelquefois par an et les intéressés peuvent y participer.

Les développeurs auront une documentation technique (une API).

Pertinence, résultats/livrables attendus et valorisation du projet

La demande peut-être être **accompagnée de tous documents utiles** :

*présentation du projet global ou descriptif du projet, rapport de phases préliminaires, étude de faisabilité (définition des besoins), dossier d'expression des besoins ou **cahier des charges**, devis détaillé…*

8.1. Résultats attendus (livrables) : (10 lignes maximum)

Une version améliorée YAO, Visual_YAO, douée d'une interface graphique qui va rendre plus accessibles la résolution des problèmes de assimilation de données.

Un manuel d'utilisation.

Un manuel du développeur.

Des documents de support de formation pour l'approche YAO de l'assimilation de données.

8.2. Pertinence du projet pour votre communauté scientifique

YAO est utilisable par la communauté des géophysiciens pour aborder des nouveaux problèmes d'assimilation des données. Pour l'utiliser il convient de définir le modèle, déclarer les modules et introduire le graphe.

Actuellement YAO est un prototype en mode texte qui a été testé avec succès sur des problèmes de taille importante. Pour l'utilisation de ce logiciel il faut avoir bien claire toute l'organisation des fichiers et la structure des noms et des extensions à donner. La création de cette structure n'est guidée que par une documentation papier.

D'autre part, les directives sont écrites d'un seul coup, en phase de déclaration du modèle direct, et s'est en exécutant que l'on relève les incohérences. L'analyse d'erreur dans une application de grande dimension se révèle très complexe (on ne sait pas où est l'erreur et sur quel attribut il faut opérer). Cette complexité peut être largement réduite si on utilise une interface graphique. Cette interface, qui contrôle la syntaxe à l'écriture, met immédiatement en évidence l'incohérence des attributs et ne permet pas la continuation de la définition de l'application.

Lors de l'écriture d'un module, la seule chose importante est de spécifier le contenu des méthodes et le nom du module mais actuellement l'utilisateur est obligé de s'occuper d'aspects plus marginaux comme la déclaration des méthodes et l'extension du fichier. Rendre automatique quelques opérations sur un problème de grands dimensions (par exemple 70 modules) peut faire économiser une quantité de temps et d'erreurs importante.

Dans l'état actuel, l'application est lancée directement à partir de la ligne de commande et les résultats affichés dans la même fenêtre sous forme de trace peu lisible. L'interface graphique devrait permettre l'exécution et la visualisation des résultats de façon claire. Enfin, l'interface graphique faciliterait grandement l'utilisation de YAO et donc la mise en oeuvre de l'assimilation variationnelle de données pour les non informaticiens, en particulier pour les membres non permanents de la JEAI/MTE.

8.3. Pertinence du projet vis à vis des objectifs de SPIRALES / justification d'un financement DSI

L'appui de la DSI constitue un apport essentiel qui rend possible cette extension de YAO.

Le résultat visé est double, d'un coté : un outil graphique facilement accessible qui permettra à un plus grand nombre de chercheurs du Sud de se mettre sans trop d'appréhension à l'assimilation des données et d'un autre : la rencontre de deux communautés scientifiques, l'informatique avec les géophysiciens qui travaillent avec l'assimilation de données.

8.4. Retours sur investissement attendus (pour l'unité, l'institut…)

Visual _YAO permettra à ses futurs utilisateurs et aux utilisateurs actuels de YAO de travailler à leurs applications de manière plus aisée. Sans nul doute, plus de géophysiciens du LTI, du LOCEAN et de la communauté des chercheurs plus largement, pourront plus facilement envisager des applications en assimilation variationnelle de données. Les retombées scientifiques en termes de nouvelles applications et en termes de publications scientifiques sont certains.

8.5. Bénéfices du projet pour les pays du Sud

L'outil sera particulièrement utile pour les laboratoires de l'UCAD et du Sénégal qui veulent se mettre à l'assimilation des données. Au sein du LTI ce projet permettra la formation pratique à l'assimilation de données et au génie logiciel des étudiants et des chercheurs. Cela permettra certainement d'ouvrir de nouvelles pistes de recherche au sein de ces laboratoires.

Plusieurs stages de master sur l'assimilation des données en océanographie et en climat seront initiés au LTI grâce à l'existence de Visual-YAO. La version actuelle de YAO n'est pas assez conviviale pour permettre son utilisation optimale par les étudiants de master 2 recherche et les doctorants de la JEAI/MTE et du LTI.

8.6. Capitalisation, valorisation, transfert de savoirs-faire ou d'outils possibles ou prévus en matière d'IS

L'outil et les applications développées seront documentés et disponibles sur le web en accès libre pour tous. La communauté y aura accès et pourra se l'approprier.

Durant le projet un transfert de savoirs-faire se fera via les formations des chercheurs et étudiants à l'assimilation variationnelle de données (d'une manière générale) et à l'utilisation de l'outil Visual_YAO.

8.7. Valorisation possible ou prévue

La valorisation du projet pourra se faire de trois manières différentes :

- Par la publication en conférence ou revue de résultats scientifiques obtenus avec l'utilisation de Visual_YAO ;
- Au travers des formations scientifiques ;
- Par l'intermédiaire du site web qui est déjà opérationnel.

8.8. Observations particulières :